Mizzou INformation and Data FUsion Lab (MINDFUL)

Designing Reliable Navigation Behaviors for Autonomous Agents in Partially Observable Grid-world Environments

Andrew R. Buck, Derek T. Anderson, James M. Keller, Cindy Bethel, and Audrey Aldridge

University of Missouri

Department of Electrical Engineering and Computer Science

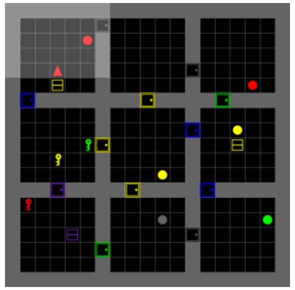
IEEE WCCI 2024 (IJCNN)

Motivation and Overview

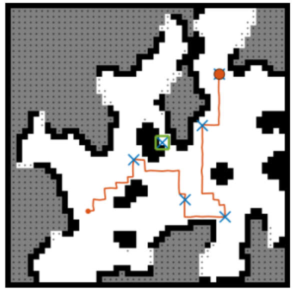
- **How does an agent decide where to go?**
  - **Needs to perform some task**
  - **Unknown environment**
  - **Explainable behavior**

- **Can we use a grid-world to study this problem?**

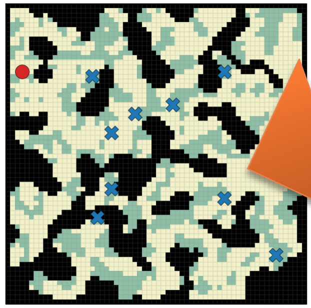
Abstract — Types of Grid-Worlds — Realistic



MiniGrid

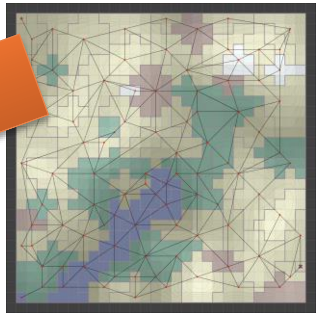


Partially Observable TSP

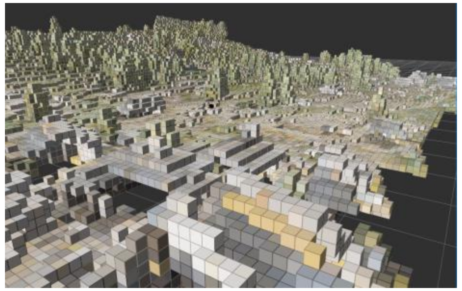


Target Detection Game

Today

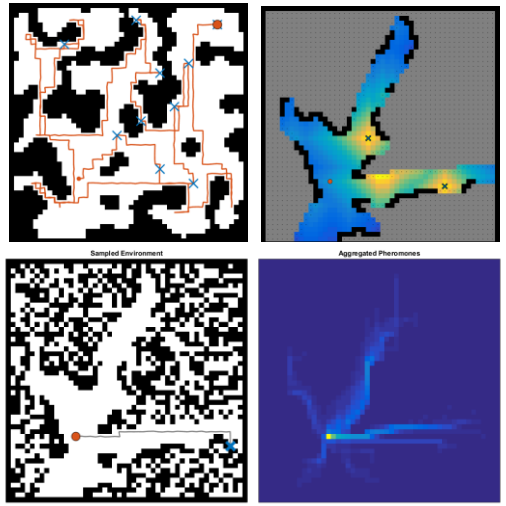


Computational Mental Map

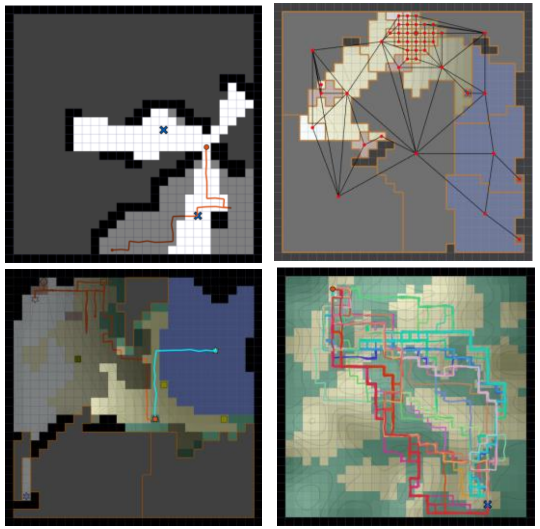


UFOMap

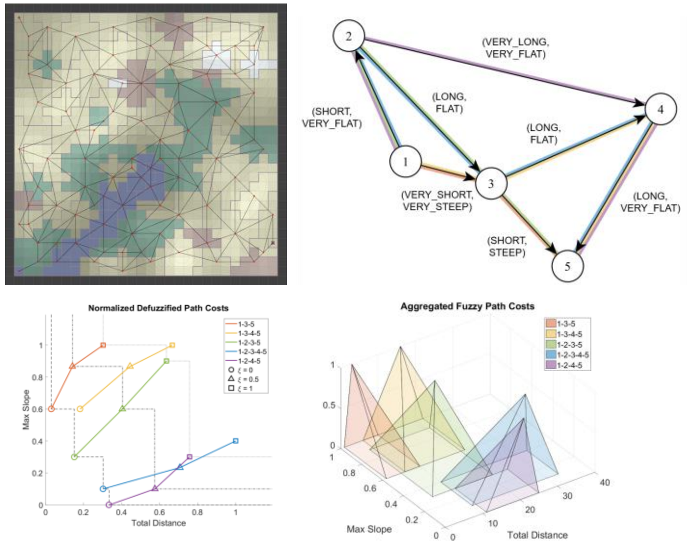
Prior Work



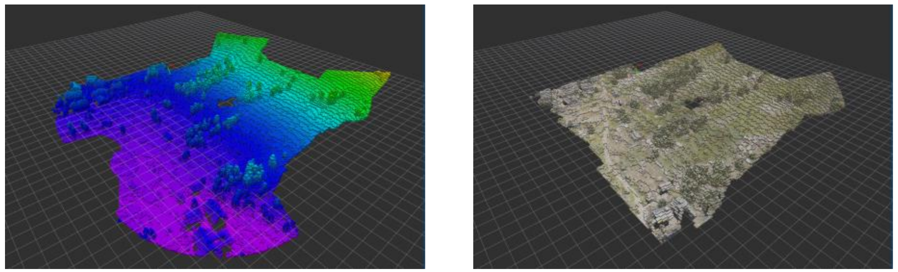
Myopic Monte-Carlo for TSP
(CEC 2016)



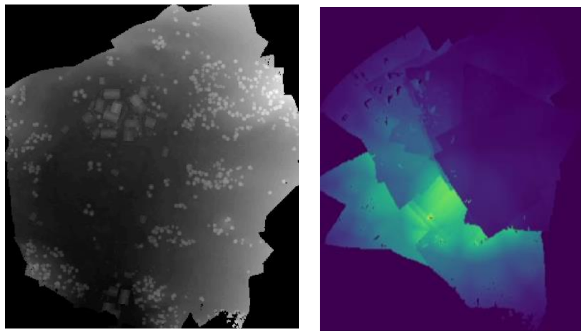
Computational Mental Map
(Ph.D. Dissertation 2018)

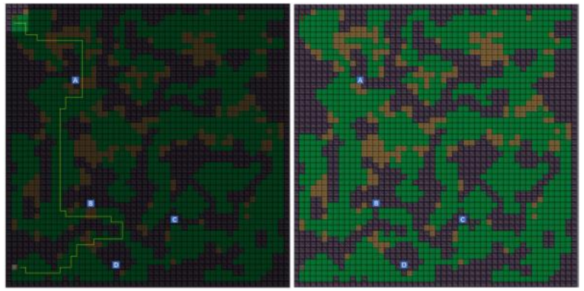


Multi-Objective Fuzzy Paths (FUZZ-IEEE 2019)



Voxel Map Feature Layers for UAV (SPIE 2022)

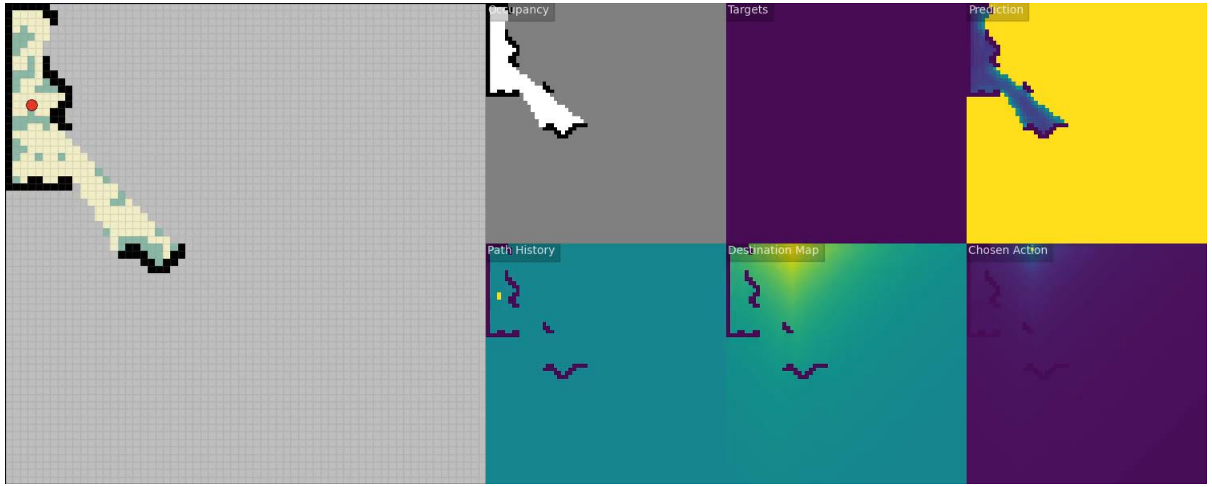




Human-Robot Teaming Game (SPIE 2023)

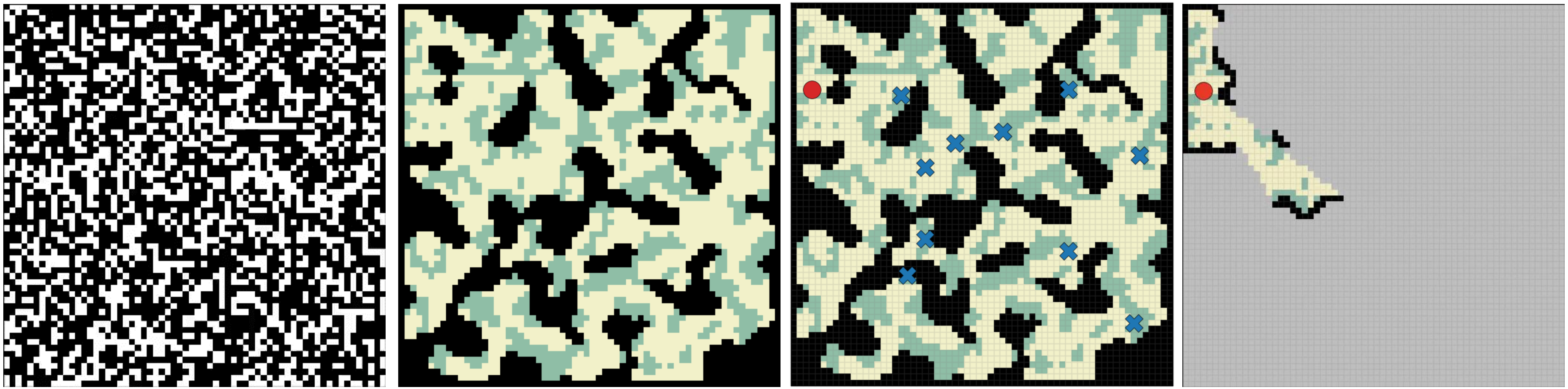
Today: Target Detection Grid-World

- **How do we build agents that can navigate reliably?**
  - We designed a custom grid-world environment to study this.
- **Can we support learning behaviors via RL?**
  - We look at how observations are mapped to actions.
- **Two approaches today:**
  - **Linear weighted policy**
    - Exploration vs. Exploitation
  - **Neural Net policy**
    - U-Net learns where to go



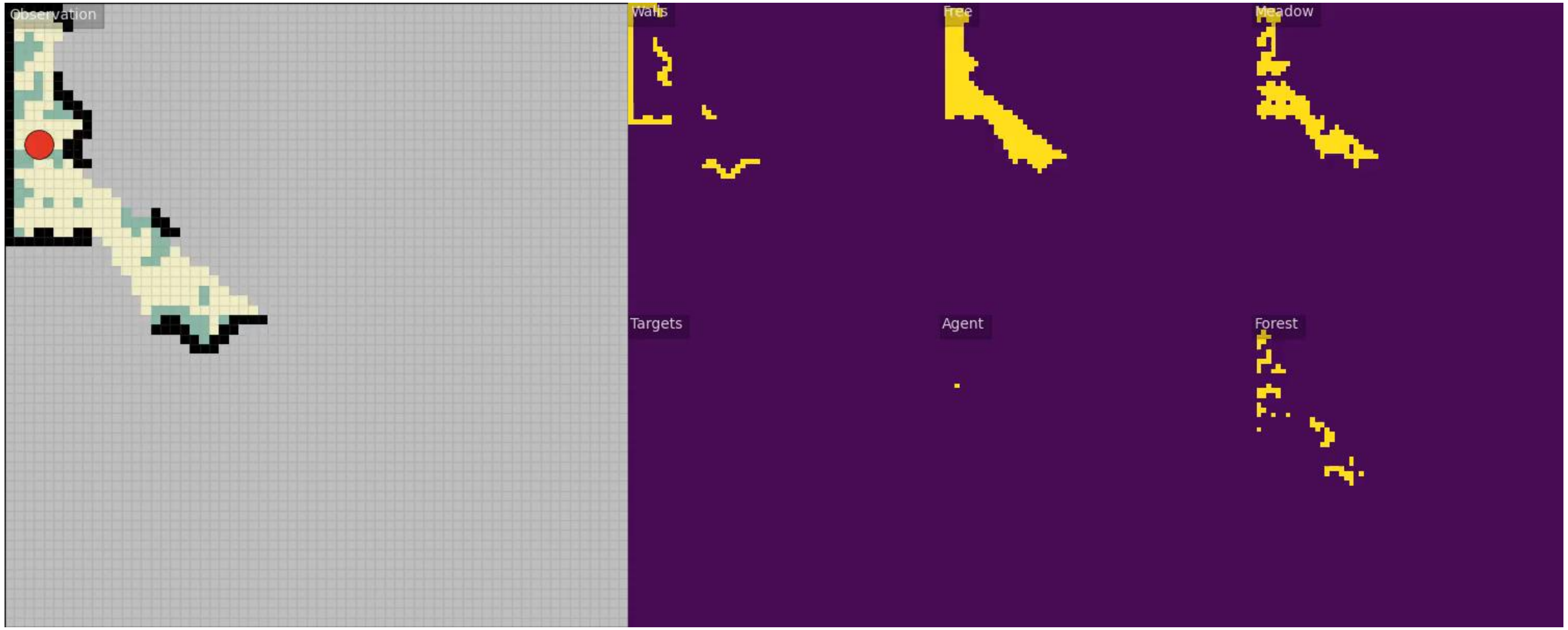
Grid-world Environment

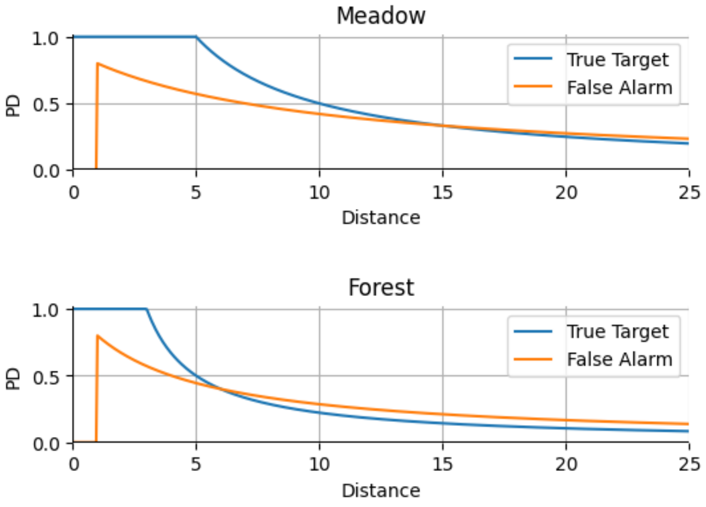
- **We used a 64x64 grid with procedurally generated features.**
  - Cellular automata used to generate walls and terrain
  - Agent and targets placed randomly
- **The agent gets a partial observation of the environment.**
  - Can move up, down, left, or right
  - Looking to find the “true target” among several “false alarms”



Observations

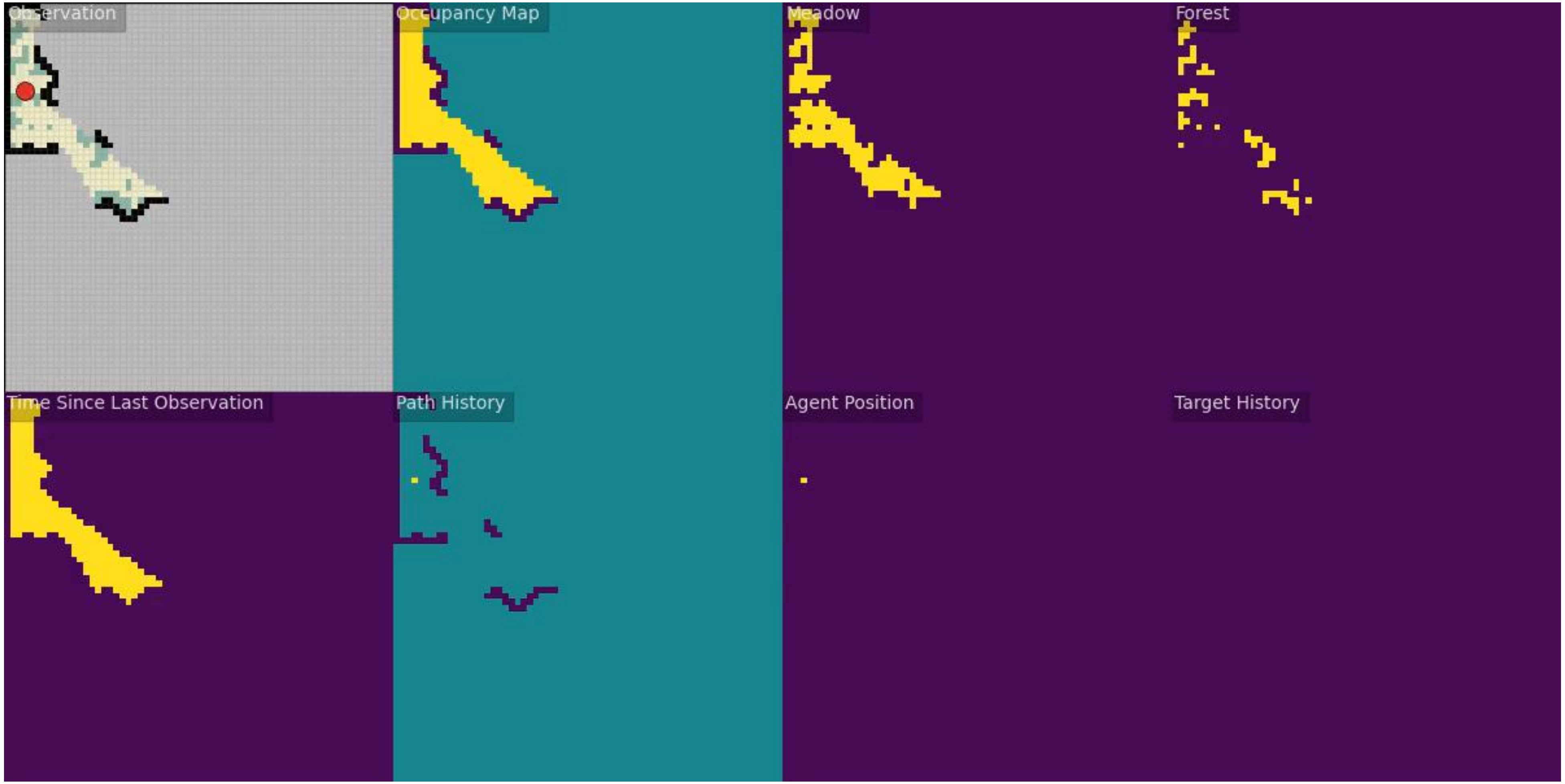
- **The agent gets a set of binary feature layers as a local observation.**
- **Target detection depends on distance and terrain type.**
  - Hard to detect at long range
  - Easier to detect in meadow than forest





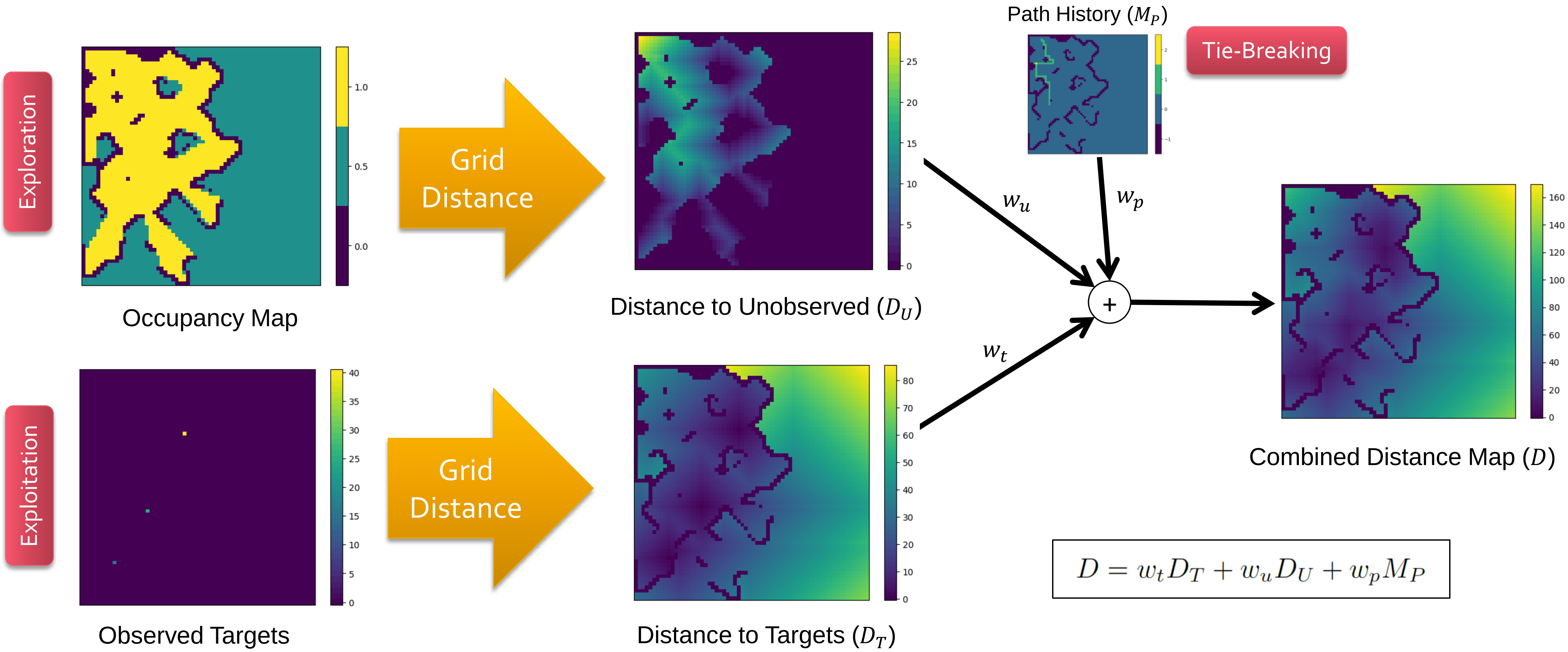
Persistent Feature Layers

- As the agent moves, the local observations are aggregated into persistent feature layers.



Multi-Criteria Decision Making

- **A linear weighted policy explicitly balances exploration and exploitation.**



$$D = w_t D_T + w_u D_U + w_p M_P$$

Exploration vs Exploitation

Exploration

$w_t = 1, \quad w_u = 2, \quad w_p = 1$

644 Steps

Balanced

$w_t = 1, \quad w_u = 1, \quad w_p = 1$

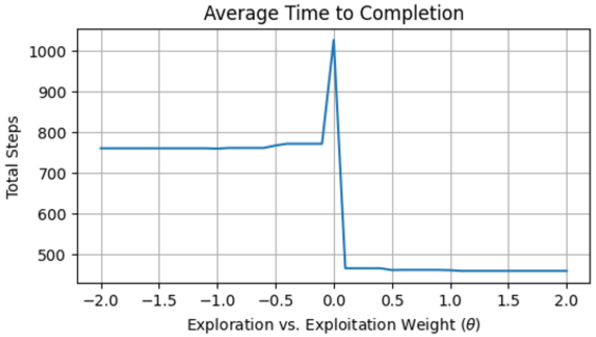
1005 Steps

Exploitation

$w_t = 2, \quad w_u = 1, \quad w_p = 1$

322 Steps

Averaged over 100 environments, exploitation consistently does best.



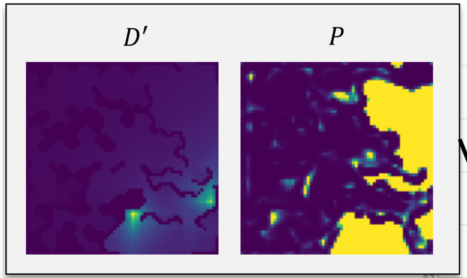
$$w_t = \begin{cases} 1+\theta, & \theta \geq 0 \\ 1, & \theta < 0 \end{cases}$$

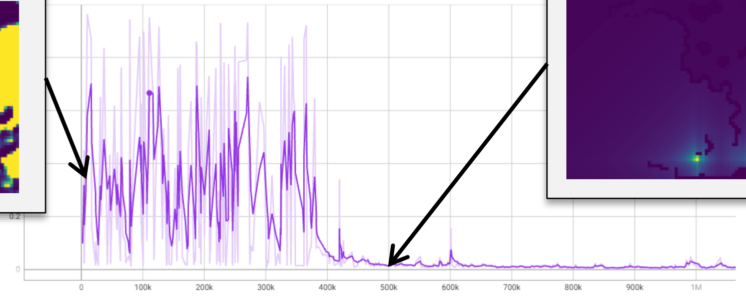
$$w_u = \begin{cases} 1, & \theta \geq 0 \\ 1-\theta, & \theta < 0 \end{cases}$$

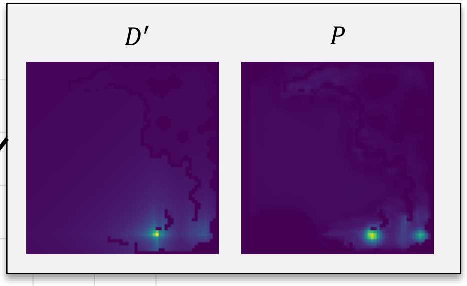
$$w_p = 1$$

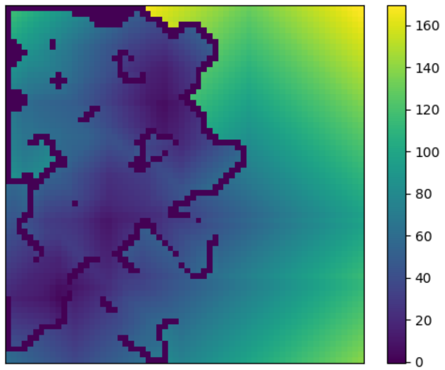
Training a U-Net

- **We train a U-Net to learn to predict the distance map.**



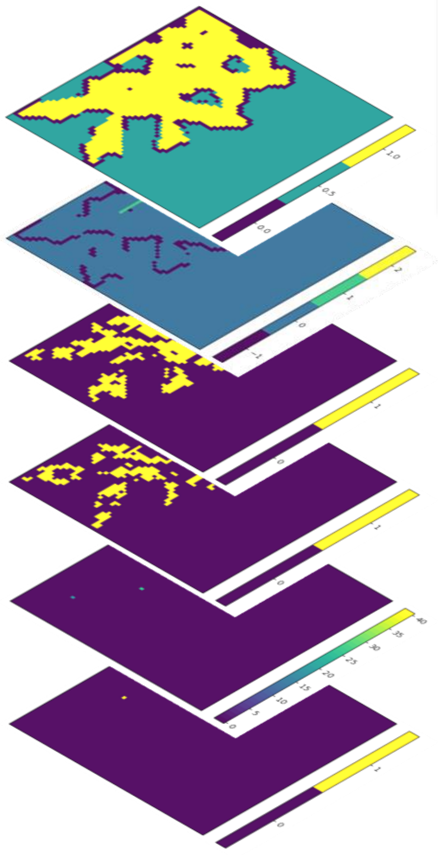




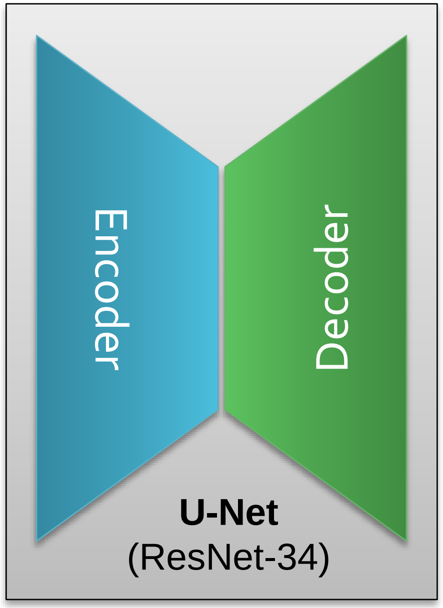


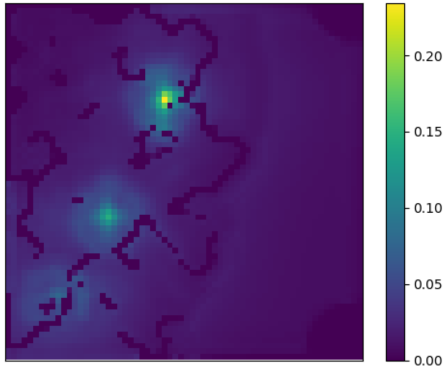
Combined Distance Map (D)

$D' = \frac{1}{D}$



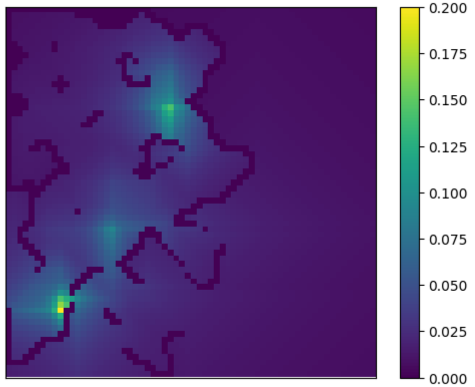
Persistent Feature Layers





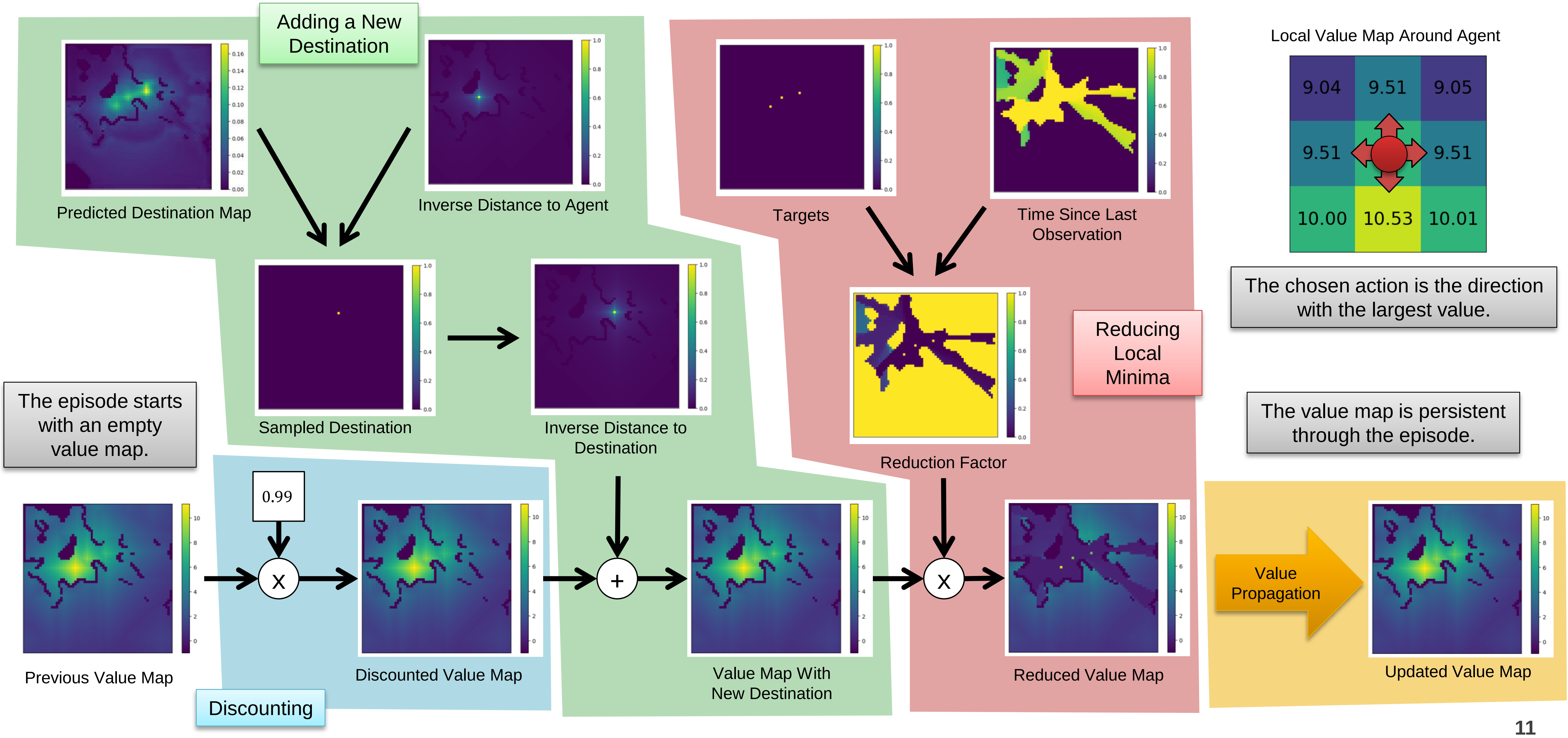
Predicted Destination Map (P)

RMSE Loss

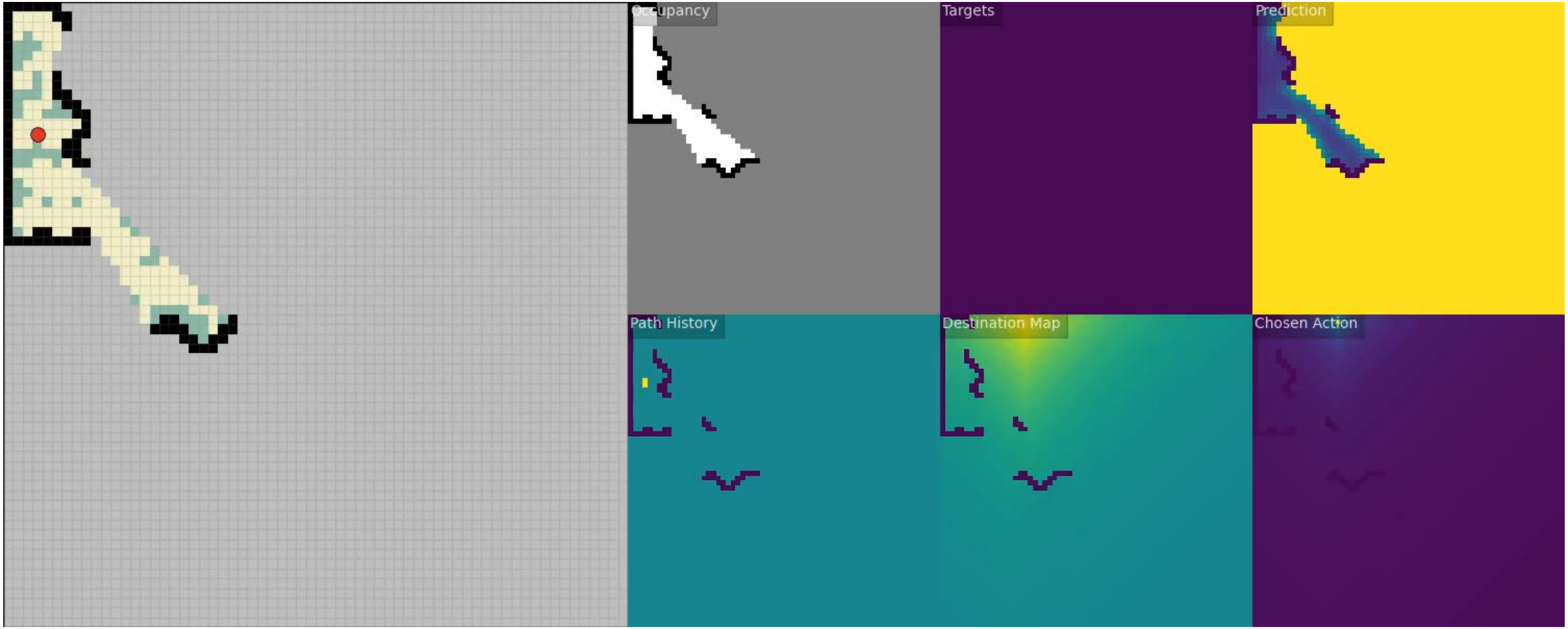


Scaled Destination Map (D')

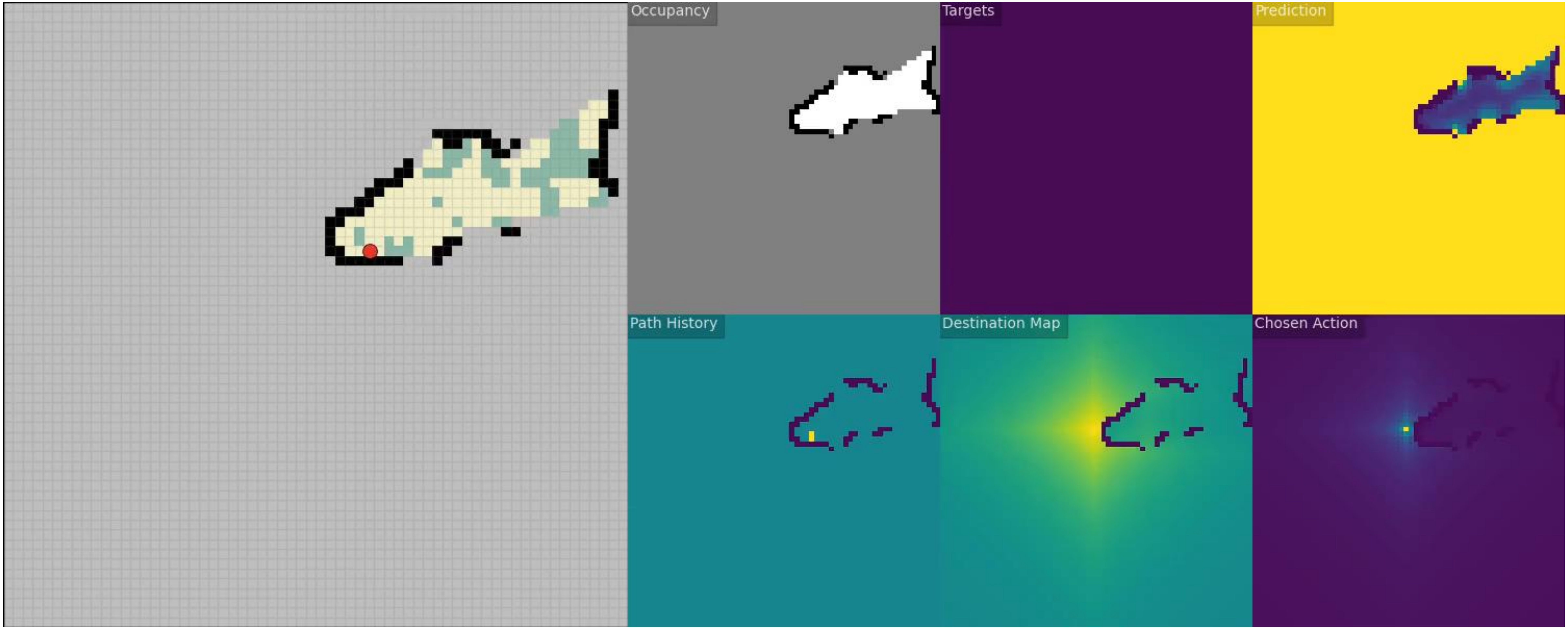
From Prediction to Action



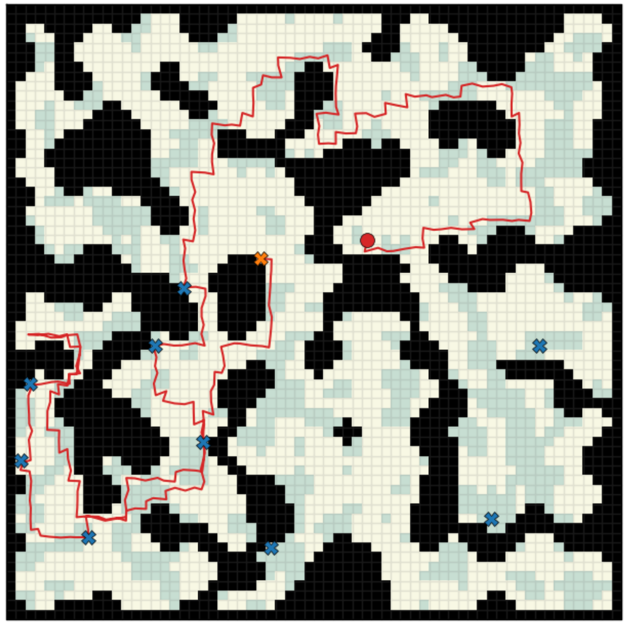
NN Policy Example 1



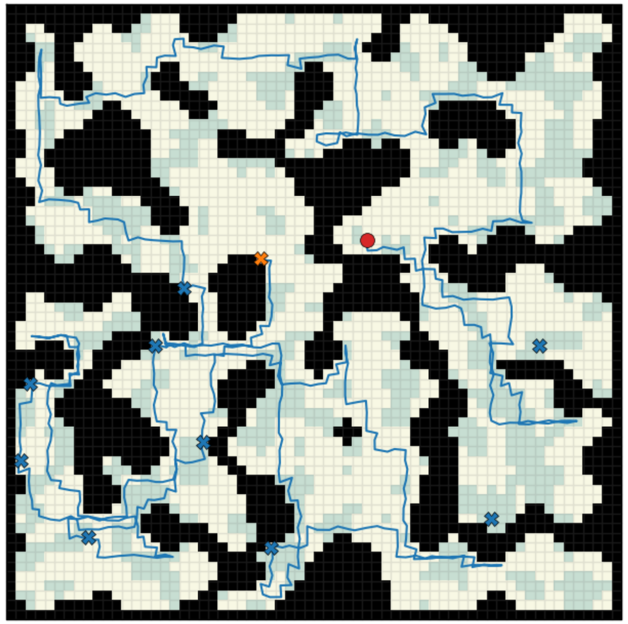
NN Policy Example 2



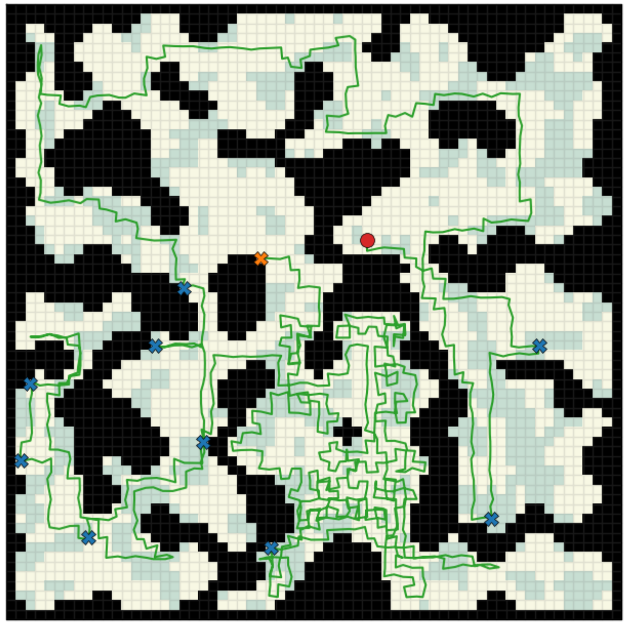
Comparing Methods



NN Policy: 282 Steps



Exploration Policy ($\theta = -1$): 578 Steps



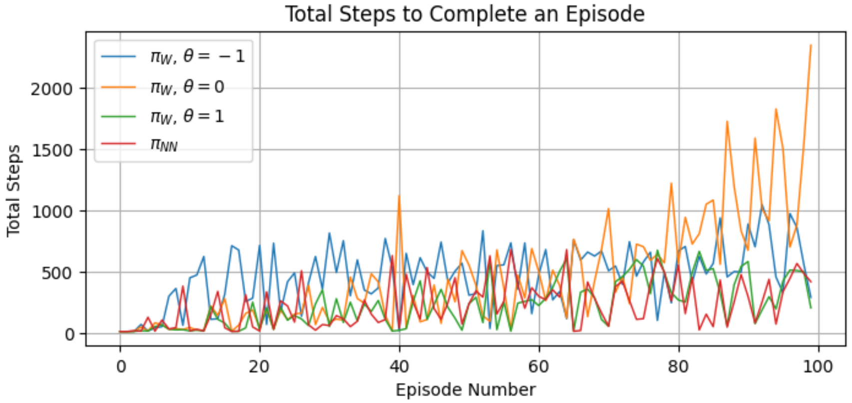
Exploitation Policy ($\theta = 1$): 880 Steps

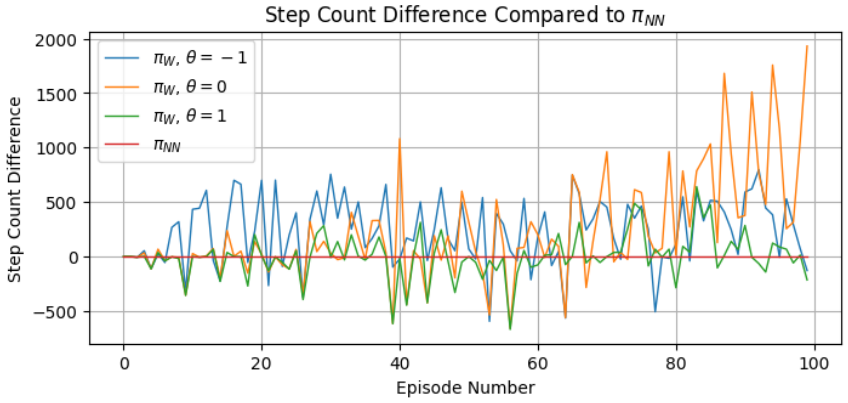
Compared methods in 100 different environments

$$w_t = \begin{cases} 1+\theta, & \theta \geq 0 \\ 1, & \theta < 0 \end{cases}$$

$$w_u = \begin{cases} 1, & \theta \geq 0 \\ 1-\theta, & \theta < 0 \end{cases}$$

$$w_p = 1$$





Conclusions and Future Work

- **We compared two types of policies for navigating in grid-world environments:**
  - A linear policy that uses manually defined features and weights
  - A NN policy that predicts where to go next
- **Both policies use a value map to direct agent actions.**
  - High-level goals are achieved through low-level actions.
- **The NN policy learns from demonstrated behavior.**
  - Environment features are saved in a “Mental Map”.
  - The network can learn to recognize desirable locations.
- **The target detection problem can be extended to more complex problems.**
  - Rewards could be tied to terrain type or other MCDM objectives.
  - Multi-agent settings can explore the potential for human-robot teaming.

Thank You!